

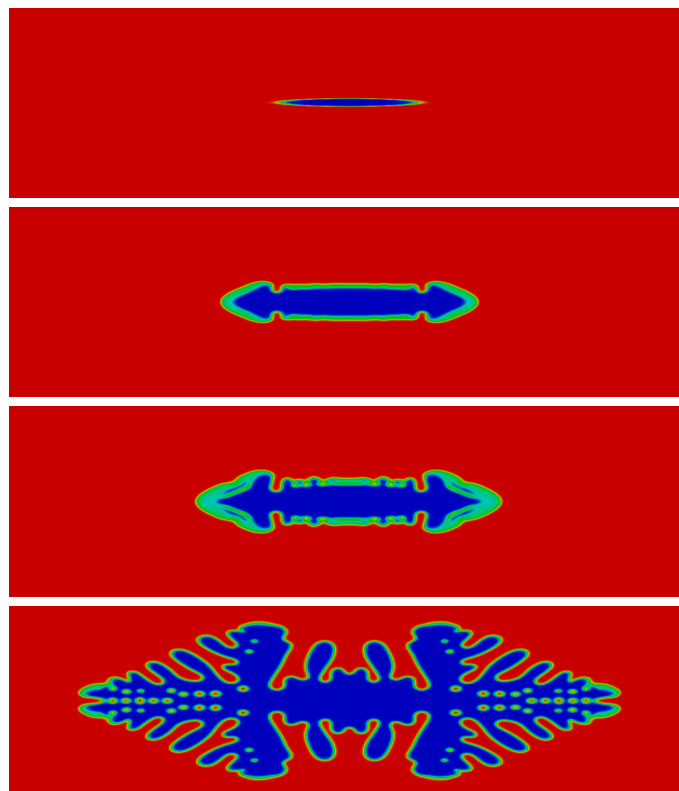
Método numérico termodinámicamente consistente de un modelo phase-field

Trabajo de fin del Máster
"Computación avanzada para ciencias e ingeniería"

Borja González Ferreiro

Tutores:

Manuel Abellanas¹ e Ignacio Romero²



Curso 2010-2011

¹Facultad de Informática

²Escuela Técnica Superior de Ingenieros Industriales de Madrid

Índice

1. Introducción	3
2. Modelo phase-field de la solidificación	4
2.1. Conservación de la energía	5
2.2. Producción entrópica positiva	6
2.3. Ecuaciones del modelo phase-field	7
2.4. Modelo I para la solidificación de Wang	8
3. Discretización	9
3.1. Discretización mediante la regla del punto medio y el método de los elementos finitos	9
3.1.1. Método de los elementos finitos	9
3.1.2. Regla del punto medio	12
3.1.3. Discretización estándar	12
3.2. Método termodinámicamente consistente	14
3.2.1. La derivada discreta	14
3.2.2. Variables del método numérico	15
4. Implementación y simulaciones numéricas	17
4.1. Implementación	17
4.1.1. IRIS, un código de elementos finitos	17
4.1.2. Paralelización del código	18
4.2. Simulaciones y resultados	20
4.2.1. Procesos de subenfriamiento	20
4.2.2. Simulaciones	22
5. Conclusión	30

Agradecimientos

En primer lugar, me gustaría agradecer a Ignacio Romero no sólo el tiempo que pacientemente me ha regalado, sino también el apoyo que me ha brindado. No ha escatimado en recursos y me ha ofrecido todos los medios que estaban a su alcance para que mi trabajo fuera lo más cómodo posible.

Debo especialmente al profesor Héctor Gómez de la Universidad de La Coruña que este trabajo tenga unas imágenes tan vistosas. Después de recibirme muy calurosamente en La Coruña, me ayudó cuando ya todo parecía perdido y gracias a él pudimos correr las simulaciones.

Con ellos, a cualquiera le entran ganas de trabajar.

1. Introducción

Es cierto que inicialmente, los modelos phase-field ayudaron a resolver modelos de solidificación. Pero, rápidamente, se amplió el espectro de problemas que se podían tratar en ciencia de materiales con estos modelos. Pronto, se convirtieron prácticamente en la herramienta preferida para simular gran cantidad de modelos de cambio de fase de primer orden, aunque no modelen un cambio de estado. Es el caso por ejemplo del cambio de la austenita a la martensita en el acero. Hoy en día, se ha extendido además su uso a problemas de otras ramas de la ciencia. Tal es el caso de la hidrología e incluso de la biología. Por ejemplo, se emplean este tipo de modelos en simulaciones de membranas celulares.

La resolución de problemas de contorno donde las interfaces son infinitesimales (“sharp boundary problems”) es muy complicada, por ello se utilizan los modelos phase field, más sencillos de resolver. En estos modelos, se define un parámetro de orden (“order parameter”), esto es en un sistema de dos fases, un campo escalar que toma un valor definido en cada fase, por ejemplo 1 en una fase, 0 en la otra y valores intermedios en la interfase. De esta forma, se pueden recuperar los problemas del tipo “sharp boundary problem”, imponiendo en el modelo phase-field que el espesor de la interfase tienda a 0.

Por otra parte, siempre que se emplean ecuaciones de este tipo para modelizar un fenómeno físico, es importante que por lo menos la termodinámica del proceso sea correcta. En el artículo [12], Penrose y Fife presentaron unas reglas que garantizan la construcción de modelos phase-field donde las dos leyes de la termodinámica se conservan. En esta formulación, la entropía se comporta como una función de Lyapunov, siendo la responsable de la cinética de los sistemas multifase.

El objetivo de este trabajo fin de máster es doble. Por una parte, queremos extender las aplicaciones de la novedosa metodología introducida en los trabajos [13, 14, 15], a sistemas acoplados que no sean mecánicos. Este nuevo marco teórico, supone una guía para encontrar discretizaciones termodinámicamente consistentes de determinado tipo de problemas. Nosotros hemos decidido estudiar un modelo de solidificación, que acopla un campo de fase con la temperatura.

Por otra parte, buscamos desarrollar el código necesario para simular un modelo concreto de solidificación. El modelo que hemos escogido es el descrito en 2.4.

Las ideas en las que nos hemos basado para desarrollar esta discretización se encuentran en los artículos [13, 14, 15]. Además de proponer una metodología general para la formulación de discretizaciones termodinámicamente consistentes de problemas de evolución, se demuestra también que los métodos numéricos que preservan la estructura termodinámica de los sistemas de ecuaciones, son muy estables (entendiendo por estabilidad que pueden ser integrados con pasos de tiempo grandes) y generan soluciones con mayor sentido físico que las obtenidas con métodos estándar.

A la hora de simular el sistema de ecuaciones, la física de la solidificación nos obliga a trabajar con mallas espaciales extremadamente finas. Por lo tanto, hemos desarrollado un código capaz de correr en sistemas paralelos, minimizando así el tiempo de ejecución.

En primer lugar, vamos a explicar en qué consisten los modelos phase-field y vamos a deducir sus ecuaciones de evolución a partir de los primeros principios

de la termodinámica. A continuación, en el apartado 2.4, vamos a explicar las funciones de estado consideradas en el trabajo de Wang et al. [17]. En la siguiente sección, presentaremos un método numérico termodinámicamente consistente para resolver el modelo. Finalmente, en la sección 4, comentaremos algunas simulaciones del modelo obtenidas con nuestro programa.

2. Modelo phase-field de la solidificación

Como se ha explicado en la introducción, los modelos de tipo phase-field se utilizan para resolver sistemas de EDPs³ con una condición de contorno algo particular. En este tipo de modelos, la condición de contorno está impuesta sobre una interfase que se mueve, a su vez, con una velocidad función del gradiente de una variable de estado. Los modelos phase-field pueden verse precisamente como una herramienta matemática que transforma un problema con frontera móvil en un sistema de EDPs, más sencillo de resolver numéricamente.

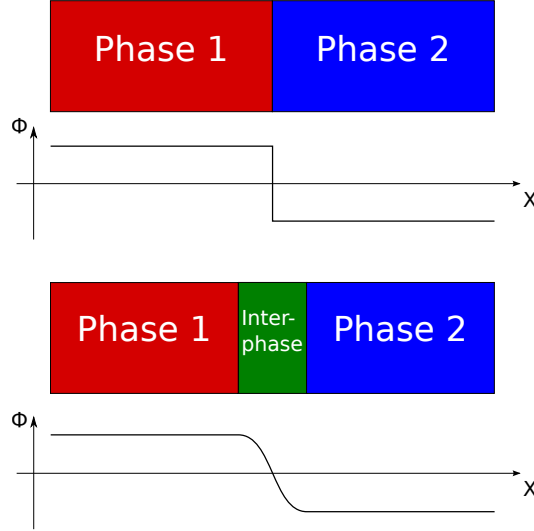


Figura 1: Ejemplo del comportamiento de un campo de fase.

La idea detrás de los modelos phase-field consiste en crear un nuevo campo ϕ , llamado campo de fase (“phase field”), continuo en el espacio pero que toma un valor constante distinto en cada fase. En la figura 1 se ha representado la evolución del nuevo campo de fase en el modelo phase-field (derecha) y la evolución que tendría en el problema inicial, que corresponde al modelo phase-field cuando el espesor de la interfase tiende a 0. La interfase se sitúa en la región donde ϕ toma un valor comprendido entre las constantes que representa cada una de las fases. En [4] se puede encontrar una introducción a los modelos phase-field.

Nuestro objetivo es resolver un sistema de ecuaciones termodinámicamente consistentes que alberguen cierto sentido físico. Históricamente, los phase-field empezaron a aplicarse al estudio de la solidificación. Entre los muchos modelos

³Ecuaciones en Derivadas Parciales.

de solidificación que componen la literatura científica, hemos trabajado con el propuesto por Wang et al. en el artículo [17] porque, como vamos a comprobar a continuación, sus ecuaciones se deducen directamente de las leyes de la termodinámica. Aunque en este artículo Wang presenta dos modelos para la solidificación de sustancias puras, nosotros hemos utilizado únicamente el modelo I.

En esta sección vamos a partir de los principios fundamentales de la termodinámica para extraer las ecuaciones del modelo de tipo phase-field que hemos resuelto y simulado numéricamente.

2.1. Conservación de la energía

Consideremos un cuerpo que ocupa un volumen V y compuesto por dos fases: una sólida y otra líquida. Para determinar qué parte del cuerpo se encuentra en fase líquida y cuál sólida a lo largo del tiempo, vamos a introducir un nuevo campo de fase ϕ , tal que $\phi = 0$ en el sólido y $\phi = 1$ en el líquido. El campo de fase es una nueva variable de estado, por lo tanto la energía y la entropía del sistema dependen de ella.

Vamos a suponer que la energía total E del sistema se puede expresar a través de su densidad e y que, el sistema, sólo puede intercambiar calor con el entorno, a través de su frontera ∂V . El primer principio de la termodinámica garantiza que:

$$\dot{E} + \int_{\partial V} \mathbf{q} \cdot \mathbf{n} \, da = 0 , \quad (1)$$

donde \mathbf{q} es el flujo de calor a través de la frontera, \mathbf{n} es su vector normal unitario hacia fuera y donde hemos recurrido a la notación $(\dot{}) = \frac{\partial}{\partial t}$. Aplicando el argumento clásico de localidad en análisis funcional, podemos afirmar que si la ecuación:

$$\int_{\partial V} (e + \mathbf{q} \cdot \mathbf{n}) \, da = 0 , \quad (2)$$

se satisface para cualquier ∂V , entonces también se cumple:

$$\dot{e} = -\nabla \cdot \mathbf{q} . \quad (3)$$

Si además suponemos que el flujo de calor \mathbf{q} puede escribirse como:

$$\mathbf{q} = M_T \nabla \left(\frac{1}{T} \right) , \quad (4)$$

siendo T la temperatura y M_T una constante positiva, la expresión (3) queda:

$$\dot{e} = -\nabla \cdot \left[M_T \nabla \left(\frac{1}{T} \right) \right] . \quad (5)$$

Esta última ecuación va a ser la primera ecuación de evolución de nuestro sistema.

2.2. Producción entrópica positiva

A partir de la conservación de la energía hemos encontrado una primera ecuación de evolución. Pero no hace intervenir el campo de fase, que es una variable de estado del problema. En los modelos phase-field, es habitual definir una energía libre dependiente del campo de fase. Penrose y Fife establecieron en [12] una serie de reglas para generar expresiones de energías libres consistentes con las leyes de la termodinámica. Más concretamente, defendieron el uso de la entropía como energía libre dependiente del campo de fase y propusieron la siguiente expresión:

$$S = \int_V \left(s - \frac{\epsilon^2}{2} |\nabla \phi|^2 \right) dv , \quad (6)$$

donde ϵ es una constante y s es la densidad entrópica que tendría el sistema si fuera monofásico. En la ecuación (6), tanto cada una de las fases como la interfase contribuyen a la entropía del sistema.

La entropía generada en un proceso se obtiene diferenciando la entropía total con respecto al tiempo. Como es práctica común en termodinámica, supongamos que la densidad de entropía se puede escribir en función de la energía y del campo de fase, i.e., $s = \tilde{s}(\phi, e)$. Entonces,

$$\begin{aligned} \dot{S} &= \int_V \left(\dot{\tilde{s}} - \epsilon^2 \nabla \dot{\phi} \cdot \nabla \phi \right) dv \\ &= \int_V \left(\frac{\partial \tilde{s}}{\partial e} \dot{e} + \left(\frac{\partial \tilde{s}}{\partial \phi} + \epsilon^2 \Delta \phi \right) \dot{\phi} - \epsilon^2 \nabla \cdot (\dot{\phi} \nabla \phi) \right) dv . \end{aligned} \quad (7)$$

Sustituyendo la expresión de \dot{e} obtenida en (5) en la ecuación anterior tenemos:

$$\begin{aligned} \dot{S} &= \int_V \left(\frac{1}{T} \nabla \cdot \mathbf{q} + \left(\frac{\partial \tilde{s}}{\partial \phi} + \epsilon^2 \Delta \phi \right) \dot{\phi} \right) dv - \int_{\partial V} \left(\epsilon^2 \dot{\phi} \nabla \phi \cdot \mathbf{n} \right) da \\ &= \int_V \left(\mathbf{q} \cdot \nabla \left(\frac{1}{T} \right) + \left(\frac{\partial \tilde{s}}{\partial \phi} + \epsilon^2 \Delta \phi \right) \dot{\phi} \right) dv - \int_{\partial V} \left(\frac{\mathbf{q}}{T} + \epsilon^2 \dot{\phi} \nabla \phi \right) \cdot \mathbf{n} da . \end{aligned} \quad (8)$$

La expresión de la producción entrópica es igual a la primera de las dos integrales que aparecen en la ecuación anterior. Efectivamente, se puede dividir la entropía en dos términos, la producción entrópica (primer sumando) y el flujo entrópico a través de la frontera ∂V (segundo sumando). El segundo principio exige que la producción entrópica sea positiva, condición que se traduce por:

$$\int_V \left(\mathbf{q} \cdot \nabla \left(\frac{1}{T} \right) + \left(\frac{\partial \tilde{s}}{\partial \phi} + \epsilon^2 \Delta \phi \right) \dot{\phi} \right) dv \geq 0 . \quad (9)$$

Podemos utilizar la ecuación anterior para definir la ley de evolución del campo de fase:

$$\tau \dot{\phi} = \frac{\partial \tilde{s}}{\partial \phi} + \epsilon^2 \Delta \phi , \quad \text{con } \tau > 0 , \quad (10)$$

de esta forma, cuando la combinamos con la ecuación (4), encontramos una producción entrópica de la forma:

$$\int_V \left(M_T |\nabla T^{-1}|^2 + \tau \dot{\phi}^2 \right) dv , \quad (11)$$

que es no negativa.

La fórmula (10) representa la evolución del campo de fase y es por lo tanto la ecuación que, junto a la expresión (5), determinan el comportamiento de nuestro sistema.

2.3. Ecuaciones del modelo phase-field

Las ecuaciones de evolución (5) y (10) son implicaciones matemáticas que hemos deducido de los dos principios fundamentales de la termodinámica. Solamente hemos supuesto dos hipótesis: el flujo de calor debía seguir una ley de Fourier generalizada, la ecuación (4), y la entropía total quedaba determinada por el funcional clásico propuesto por Penrose y Fife [12] muy utilizado en los problemas phase-field. Por lo demás, las ecuaciones de evolución son completamente generales y válidas para modelizar multitud de sistemas. Pero para caracterizar el sistema concreto que nosotros vamos a estudiar nos hace entonces falta más información. Debemos encontrar un modelo de las densidades de energía y de entropía. Utilizando los desarrollos descritos en [17], vamos a conseguir todo lo que necesitamos para completar la formulación de nuestro problema de valores iniciales.

En primer lugar, concentrémonos en la densidad de energía. Vamos a suponer que puede expresarse como:

$$e = \hat{e}(\phi, T) = e_L(T) + [p(\phi) - 1]L(T) , \quad (12)$$

donde e_L es la densidad de energía interna del líquido, $p(\phi)$ es un polinomio que satisface $p(0) = 0$ y $p(1) = 1$ y $L(T) = e_L(T) - e_S(T)$ es la variación entre las densidades de energía del líquido y del sólido. Además, la energía interna en el líquido e_L es de la forma

$$e_L(T) = e_L(T_M) + c(T - T_M) , \quad (13)$$

en donde c es la capacidad calorífica por unidad de volumen del líquido y T_M es la temperatura de solidificación.

Las dos nuevas ecuaciones que acabamos de introducir, (12) y (13), nos han permitido encontrar la densidad de energía interna. Pero aún tenemos que hallar una expresión correcta para la entropía. Vamos a trabajar con la densidad de energía libre de Helmholtz, definida como la transformación de Legendre de la energía interna e que sustituye la entropía por la temperatura como variable independiente.

$$\begin{aligned} f &= f(\phi, T) = e(\phi, s) - Ts \\ \Rightarrow df &= \frac{\partial f}{\partial T} dT + \frac{\partial f}{\partial \phi} d\phi . \end{aligned} \quad (14)$$

Dividiendo f por T ,

$$\frac{\partial (f/T)}{\partial \phi} = -\frac{e}{T^2} , \quad (15)$$

que tras integración a ϕ constante se convierte en:

$$f(\phi, T) = T \left(\int_T^{T_M} \frac{e(\phi, \psi)}{\psi^2} d\psi + \frac{1}{4a} g(\phi) \right) , \quad (16)$$

donde a es una constante y $g(\phi)$ un potencial de doble pozo con dos mínimos, uno en 0 y otro en 1. Por su simplicidad, vamos a escoger como potencial:

$$g(\phi) = \phi^2(1 - \phi)^2 . \quad (17)$$

Además, podemos expresar la energía libre de Helmholtz (14) en función de e_L en lugar de e apoyándonos en la ecuación (12):

$$f(\phi, T) = T \left(- \int_{T_M}^T \frac{e_L(\psi, \phi)}{\psi^2} d\psi - [p(\phi) - 1]Q(T) + \frac{1}{4a}g(\phi) \right) , \quad (18)$$

con Q definido por

$$Q(T) = \int_{T_M}^T \frac{L(\psi)}{\psi^2} d\psi . \quad (19)$$

La primera integral puede calcularse a partir de (13):

$$\begin{aligned} \int_{T_M}^T \frac{e_L(\psi, \phi)}{\psi^2} d\psi &= \int_{T_M}^T \frac{e_L(T_M) - cT_M}{\psi^2} d\psi + \int_{T_M}^T \frac{c\psi}{\psi^2} d\psi \\ &= -[e_L(T_M) - cT_M] \left(\frac{1}{T} - \frac{1}{T_M} \right) + c \ln \left(\frac{T}{T_M} \right) . \end{aligned} \quad (20)$$

Con las ecuaciones (18), (19) y (20) la energía libre de Helmholtz queda totalmente determinada. Ahora que disponemos de una expresión de f y de e , podemos deducir la forma que tiene la densidad de entropía simplemente recordando la definición de f (14):

$$s(\phi, T) = \frac{e - f(\phi, T)}{T} . \quad (21)$$

2.4. Modelo I para la solidificación de Wang

La meta que perseguíamos, en el subapartado anterior, era determinar por completo el comportamiento físico del sistema que vamos a estudiar. Por ello, hemos introducido algunas restricciones sobre la energía y la entropía. Aunque ya casi lo hemos logrado, aún nos quedan dos parámetros por determinar: $L(T)$ y $p(\phi)$. En [17], Wang propone dos modelos y cada uno de ellos otorga expresiones distintas a $L(T)$ y $p(\phi)$. Nosotros hemos implementado el modelo I.

En este modelo en concreto $L(T)$ es una constante llamada calor latente de fusión.

$$L(T) = L(T_M) = L_0 \quad (22)$$

y el polinomio $p(\phi)$ tiene la forma:

$$p(\phi) = \phi^3 (10 - 15\phi + 6\phi^2) \quad (23)$$

Al empezar el subapartado anterior, impusimos que la energía interna tuviese la forma (12), donde p tenía dos restricciones: $p(0) = 0$ y $p(1) = 1$. Podemos comprobar fácilmente que la expresión (23) efectivamente satisface sendas restricciones.

Las ecuaciones (5) y (10) rigen la solidificación de sustancias puras y garantizan tanto el crecimiento de la entropía, como la conservación de la energía.

Sin embargo, necesitan ser completadas con una expresión de las densidades de energía interna y de entropía. Por una parte, la ecuación (12) junto con la (13) caracterizan la densidad de la energía interna. Por otra parte, las ecuaciones (18), (19), (20) y (21) constituyen un modelo satisfactorio de la entropía junto con las ecuaciones del modelo I de Wang (22) y (23).

En el siguiente apartado, vamos a discretizar las ecuaciones de evolución (5) y (10) en el tiempo y en el espacio de tal forma que, en cada paso de tiempo, se conserven el primer y segundo principio de la termodinámica.

3. Discretización

En este apartado explicamos una discretización de las ecuaciones (5) y (10) que gobiernan el modelo phase-field, propuesto para modelizar la solidificación en la sección (2). La propiedad más importante que va a poseer esta discretización es la satisfacción de los dos principios de la termodinámica. Antes de presentar este nuevo método numérico, vamos a explicar en qué consiste la discretización estándar mediante el método de los elementos finitos en el espacio y la regla del punto medio en el tiempo. Nos apoyaremos en este método estándar para construir nuestra propia discretización y como comparación.

3.1. Discretización mediante la regla del punto medio y el método de los elementos finitos

Antes de discretizar el sistema de ecuaciones, vamos a explicar brevemente en qué consisten el método de los elementos finitos y la regla del punto medio.

3.1.1. Método de los elementos finitos

El método de los elementos finitos es un algoritmo, ampliamente empleado en ingeniería, para resolver numéricamente sistemas de ecuaciones en derivadas parciales, transformando el sistema de ecuaciones diferenciales en un sistema algebraico.

La resolución por el método de elementos finitos suele dividirse en cuatro etapas:

- Formulación del sistema de EDPs en forma variacional.
- División del dominio espacial de las variables de estado en subdominios llamados elementos finitos. Esto se consigue con un mallado y se crea un espacio vectorial de dimensión finita. La solución final será una combinación lineal de vectores de una base de ese nuevo espacio vectorial.
- Proyección del problema variacional sobre el espacio vectorial de los elementos finitos. De esta forma obtenemos un sistema lineal de dimensión finita.
- Resolución del sistema de ecuaciones mediante técnicas de álgebra lineal.

Vamos a ilustrar cada uno de estos pasos a través de un ejemplo, la resolución de la ecuación de Poisson con condiciones de contorno de tipo Neumann y Dirichlet. Buscamos el campo u , perteneciente a un espacio de Banach y definido

sobre un abierto $\Omega \subset \mathbb{R}^n$ con una frontera suave a trozos. Además, u satisface la ecuación de Poisson:

$$\begin{aligned}\Delta u + f &= 0, \text{ en } \Omega \\ u &= g, \text{ sobre } \Gamma_g \\ \nabla u \cdot \mathbf{n} &= h, \text{ sobre } \Gamma_h,\end{aligned}\tag{24}$$

donde Γ_h y Γ_g son abiertos de Γ , la frontera de Ω . Ahora necesitamos definir el espacio de funciones de prueba \mathcal{S} y el espacio \mathcal{V} , subespacios del espacio de Hilbert H_1 ⁴ tales que:

$$\mathcal{S} = \{u | u \in H_1 \text{ y } u = g \text{ en } \Gamma_g\}\tag{25}$$

$$\mathcal{V} = \{w | w \in H_1 \text{ y } w = 0 \text{ en } \Gamma_h\}.\tag{26}$$

Formulación débil La formulación débil (o formulación variacional) de un sistema de ecuaciones diferenciales consiste en transformar el sistema en otro de ecuaciones integrales, de tal forma que se puede resolver mediante las técnicas del álgebra lineal sobre espacios funcionales. Con la notación introducida en el párrafo anterior, podemos reescribir el problema de Poisson (24):

$$\int_{\Omega} \nabla u \cdot \nabla w \, d\Omega = \int_{\Omega} w f \, d\Omega + \int_{\Gamma_h} w h \, d\Gamma_h, \text{ con } u \in \mathcal{S} \text{ y } w \in \mathcal{V}.\tag{27}$$

Es sencillo deducir la forma débil a partir del problema inicial (24) (llamado forma fuerte):

$$\begin{aligned}0 &= \Delta u + f, \text{ en } \Omega \\ &= \int_{\Omega} w (\Delta u + f), \text{ integrando por partes} \\ &= - \int_{\Omega} \nabla u \cdot \nabla w \, d\Omega + \int_{\Gamma} w \nabla u \cdot \mathbf{n} \, d\Gamma + \int_{\Omega} w f \, d\Omega,\end{aligned}$$

imponiendo $w = 0$ en Γ_h y $\nabla u \cdot \mathbf{n} = h$, en Γ_h :

$$\begin{aligned}0 &= - \int_{\Omega} \nabla u \cdot \nabla w \, d\Omega + \int_{\Gamma_h} w h \, d\Gamma_h + \int_{\Omega} w f \, d\Omega \\ \int_{\Omega} \nabla u \cdot \nabla w \, d\Omega &= \int_{\Omega} w f \, d\Omega + \int_{\Gamma_h} w h \, d\Gamma_h, \text{ con } u \in \mathcal{S} \text{ y } w \in \mathcal{V}.\end{aligned}$$

La demostración de la implicación inversa, esto es, la forma débil es condición suficiente para que la forma fuerte se cumpla, se puede encontrar en [6]. Vamos a introducir la siguiente notación que utilizaremos de ahora en adelante:

$$\begin{aligned}\langle \nabla w, \nabla u \rangle &= \int_{\Omega} \nabla u \cdot \nabla w \, d\Omega \\ \langle w, f \rangle &= \int_{\Omega} w f \, d\Omega \\ \langle w, h \rangle_h &= \int_{\Gamma_h} w h \, d\Gamma_h.\end{aligned}$$

Con esta nueva notación la forma débil se puede reescribir de forma mucho más reducida:

$$\langle \nabla w, \nabla u \rangle = \langle w, f \rangle + \langle w, h \rangle_h\tag{28}$$

⁴ H_1 es el espacio de funciones de cuadrado integrable cuya derivada también es de cuadrado integrable. El espacio de funciones de cuadrado integrable se denota L^2

Formulación de Galerkin Hasta ahora hemos transformado una EDP en una ecuación integral. En lugar de resolver esos sistemas directamente, vamos a aproximarlos por un sistema lineal de ecuaciones.

En primer lugar, vamos a particionar el conjunto Ω , sobre el que está definido el campo u que buscamos, en n subdominios $\{\Omega^{(e)}\}_{e=1}^n$. La intersección de dos subdominios cualesquiera debe ser nula y la unión de todos los subdominios debe formar Ω . Se llama elemento a cada $\Omega^{(h)}$. En definitiva, hemos mallado Ω y obtenido una malla formada por los elementos $\Omega^{(h)}$ y m nodos de coordenadas x_A en cada elemento.

A continuación se definen dos espacios funcionales \mathcal{S}^h y \mathcal{V}^h de dimensión finita, sobre cada elemento. Estos espacios son aproximaciones de dimensión finita de \mathcal{S} y \mathcal{V} . Asumimos que cada miembro de \mathcal{S}^h puede representarse mediante la relación:

$$u^h = v^h + g^h, \text{ donde } v^h \in \mathcal{V}^h \text{ y } g^h = u^h \text{ en } \Gamma_g$$

A partir de la forma débil del problema de Poisson podemos estudiar lo que se conoce como formulación de Galerkin:

$$\langle \nabla w^h, \nabla u^h \rangle = \langle w^h, f \rangle + \langle w^h, h \rangle_h - \langle \nabla w^h, \nabla g^h \rangle_g. \quad (29)$$

Necesitamos construir una base para cada espacio \mathcal{V}^h . A los elementos de dicha base se les llama funciones de forma. Como base, se elige la familia de polinomios de grado inferior o igual a m . Cada elemento $w^h \in \mathcal{V}^h$ puede expresarse entonces como:

$$w^h = \sum_{A=1}^m N_A(x) w^h(x_A),$$

donde N_A son las funciones de forma. Además:

$$\begin{aligned} w^h &= \sum_{A=1}^m N_A(x) w^h(x_A) \\ v^h &= \sum_{A=1}^m N_A(x) v^h(x_A) \\ g^h &= \sum_{A=1}^{m_g} N_A(x) g^h(x_A) \end{aligned}$$

Las funciones de forma se escogen de tal forma que $N_A(x_B) = 0$ cuando x_B es la posición de un nodo diferente de A , mientras que $N_A(x_A) = 1$. Las funciones de forma son por lo tanto funciones de interpolación polinómica. Con esta notación, la formulación de Galerkin queda:

$$\sum_{B=1}^n \langle \nabla N_A, \nabla N_B \rangle d_B = \langle N_A, f \rangle + \langle N_A, h \rangle_h - \sum_{B=1}^{n_g} \langle \nabla N_A, \nabla N_B \rangle \quad (30)$$

para cualquier nodo A y llamando d_B a $v^h(x_B)$. La ecuación anterior esconde en realidad un sistema lineal de ecuaciones que puede ser reformulado en forma matricial. Si llamamos:

$$\begin{aligned} F_A &= \langle N_A, f \rangle + \langle N_A, h \rangle_h - \sum_{B=1}^{n_g} \langle \nabla N_A, \nabla N_B \rangle \\ K_{AB} &= \langle \nabla N_A, \nabla N_B \rangle, \end{aligned}$$

podemos construir la matriz \mathbf{K} ensamblando los términos \mathbf{K}_{AB} : $K(i, j) = K_{ij}$. De manera similar, construimos los vectores \mathbf{d} y \mathbf{F} : $\mathbf{d}(i) = d_i$, $\mathbf{F}(i) = F_i$. Finalmente, la ecuación (30) se expresa en forma matricial como:

$$\mathbf{Kd} = \mathbf{F}. \quad (31)$$

3.1.2. Regla del punto medio

Al hacer uso del método de los elementos finitos para discretizar en el espacio las ecuaciones de evolución, podemos resolver para un instante dado el modelo phase-field. Sin embargo, en las ecuaciones aparecen derivadas con respecto al tiempo que debemos discretizar. Vamos a crear un nuevo método basado en la regla del punto medio. Repasemos por lo tanto rápidamente las propiedades más importantes del método de derivación numérica del punto medio.

Un forma de encontrar fórmulas de derivadas con cualquier orden de precisión consiste en recurrir a la interpolación de Lagrange. Si $\{x_0, x_1, \dots, x_n\}$ son $n+1$ puntos distintos de Ω y $f \in C^{n+1}(\Omega)$, entonces la fórmula de interpolación polinómica de Lagrange establece la igualdad:

$$f(x) = \sum_{k=0}^n f(x_k) l_k(x) + \prod_{k=0}^n \frac{(x - x_k)}{(n+1)!} f^{(n+1)}(\xi(x)) \quad (32)$$

donde l_k es el k -ésimo polinomio de Lagrange de f en x_0, x_1, \dots, x_n .

Diferenciando la expresión anterior llegamos a:

$$f'(x) = \sum_{k=0}^n f(x_k) l'_k(x) + \frac{d \prod_{k=0}^n \frac{(x - x_k)}{(n+1)!}}{dx} f^{(n+1)}(\xi(x)) + \prod_{k=0}^n \frac{(x - x_k)}{(n+1)!} \frac{df^{(n+1)}(\xi(x))}{dx} \quad (33)$$

y evaluando la derivada en el punto x_j :

$$f'(x_j) = \sum_{k=0}^n f(x_k) l'_k(x_j) + \prod_{k=0}^n (x_j - x_k) \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \quad (34)$$

Si ahora consideramos solamente tres puntos: $x_0 - h$, x_0 y $x_0 + h$, la formula anterior queda:

$$\begin{aligned} f'(x_0) &= \frac{1}{2h} (f(x_0 + h) - f(x_0 - h)) - \frac{h^2}{6} f^{(3)}(\xi(x_0)) \\ &= \frac{1}{2h} (f(x_0 + h) - f(x_0 - h)) + o(h^2) \end{aligned} \quad (35)$$

La ecuación (35) nos da la fórmula que vamos a utilizar para aproximar numéricamente la derivada en función del tiempo en el método estándar. Comprobamos que la aproximación es de segundo orden

3.1.3. Discretización estándar

En el modelo presentado en el apartado (2) hay dos campos independientes ϕ y T . En lugar de utilizar la temperatura, vamos a trabajar con su inverso $\omega = T^{-1}$ por simplicidad de cálculo. Todas las demás magnitudes que hemos

introducido hasta ahora en este trabajo pueden expresarse en función de esas dos variables ϕ y ω . En particular las densidades de energía interna y de entropía:

$$e = \hat{e}(\phi, \omega) , \quad s = \hat{s}(\phi, \omega) . \quad (36)$$

En termodinámica suele considerarse la energía interna o la entropía como variable de estado, por lo que también podemos expresar la entropía en función de la energía interna:

$$s = \tilde{s}(\phi, e) . \quad (37)$$

Debemos tener especial cuidado cuando manipulemos la entropía, porque hemos definido dos expresiones funcionales distintas de la misma magnitud física. A pesar de que $\hat{s}(\phi, \omega)$ y $\tilde{s}(\phi, e)$ representan ambas a la entropía y su espacio imagen es el mismo, dependen de variables distintas y, por lo tanto, están definidas en espacios diferentes. Así:

$$\begin{aligned} \frac{\partial \hat{s}}{\partial \phi} &\neq \frac{\partial \tilde{s}}{\partial \phi} , \\ \frac{\partial \hat{s}}{\partial e} &= 0 , \quad \frac{\partial \tilde{s}}{\partial \omega} = 0 . \end{aligned}$$

En primer lugar, vamos a obtener la forma débil de las ecuaciones (5) y (10). Si ϕ y ω pertenecen al espacio de Hilbert $H^1(V)$,⁵ entonces para cualesquiera ψ y μ también en $H^1(V)$, se deben satisfacer las siguientes ecuaciones variacionales:

$$\begin{aligned} \left\langle \frac{\partial \hat{e}}{\partial \phi} \dot{\phi} + \frac{\partial \hat{e}}{\partial \omega} \dot{\omega}, \psi \right\rangle &= \langle M_T \nabla \omega, \nabla \psi \rangle , \\ \left\langle \tau \dot{\phi}, \mu \right\rangle &= \left\langle \frac{\partial \hat{s}}{\partial \phi}, \mu \right\rangle - \langle \epsilon^2 \nabla \phi, \nabla \mu \rangle . \end{aligned} \quad (38)$$

Para formular la discretización de las ecuaciones anteriores, empezamos definiendo una partición del intervalo de tiempo $[0, T)$ en N intervalos disjuntos $[t_n, t_{n+1})$ con $0 \leq n \leq N$ y $0 = t_0 < t_1 < \dots < t_N = T$.

Para discretizar en el espacio, vamos a considerar una triangulación del volumen V en elementos V_e conectados a N_{nd} nodos de coordenadas $x_a, a = 1, \dots, N_{nd}$. Utilizando las funciones de forma estándar de los elementos finitos $N^a : V \rightarrow \mathbb{R}$, definimos un subconjunto de $H^1(V)$ de dimensión finita:

$$\mathcal{V}^h = \left\{ u^h \in H^1(V), \quad u^h = \sum_{a=1}^{N_{nd}} N_a(x) u^a \right\} . \quad (39)$$

La discretización estándar que estábamos buscando aplicando la regla del punto medio en el tiempo y elementos finitos en el espacio es el par $(\phi_{n+1}^h, \omega_{n+1}^h) \in \mathcal{V}^h \times \mathcal{V}^h$, que, dados (ϕ_n^h, ω_n^h) satisface:

$$\begin{aligned} \left\langle \frac{\partial \hat{e}}{\partial \phi} \frac{\phi_{n+1}^h - \phi_n^h}{\Delta t} + \frac{\partial \hat{e}}{\partial \omega} \frac{\omega_{n+1}^h - \omega_n^h}{\Delta t}, \psi^h \right\rangle &= \left\langle M_T \nabla \omega_{n+\frac{1}{2}}^h, \nabla \psi^h \right\rangle , \\ \left\langle \tau \frac{\phi_{n+1}^h - \phi_n^h}{\Delta t}, \mu^h \right\rangle &= \left\langle \frac{\partial \hat{s}}{\partial \phi}, \mu^h \right\rangle - \left\langle \epsilon^2 \nabla \phi_{n+\frac{1}{2}}^h, \nabla \mu^h \right\rangle . \end{aligned} \quad (40)$$

⁵ $H^1(V)$ es el conjunto de funciones de cuadrado integrable cuyas derivadas parciales son también de cuadrado integrable.

para cualesquiera $(\psi^h, \mu^h) \in \mathcal{V}^h \times \mathcal{V}^h$. Todas la derivadas parciales de la energía interna y de la entropía presentes en el ecuación anterior deben evaluarse en el punto medio: $(\phi_{n+\frac{1}{2}}^h, \omega_{n+\frac{1}{2}}^h) = \frac{1}{2} [(\phi_n^h, \omega_n^h) + (\phi_{n+1}^h, \omega_{n+1}^h)]$.

3.2. Método termodinámicamente consistente

El método (40) tiene una precisión de segundo orden en el tiempo pero genera soluciones discretas $\{(\phi_n^h, \omega_n^h)\}_{n=0}^N$ cuya energía puede no conservarse y cuya entropía puede decrecer entre instantes de tiempo distintos. El objetivo principal de este trabajo es modificar este método numérico para remediar esas inconsistencias con las leyes físicas.

Para ello vamos a utilizar una derivada numérica diferente a la del punto medio. Además vamos a modificar las variables del método numérico que vamos a ir calculando en cada paso de tiempo.

3.2.1. La derivada discreta

El concepto de derivada discreta (o gradiente discreto) es utilizado habitualmente en la formulación de métodos conservativos (ver, por ejemplo, [3, 11, 13]). La idea, que se remonta por lo menos a [8, 9, 5, 7], consiste en definir una construcción explícita, con el fin de obtener una aproximación de segundo orden de funciones gradiente generales que satisfagan la condición de direccionalidad (42), explicada más abajo.

Se puede construir la derivada discreta de funciones reales definidas en la recta real, en campos de vectores o incluso en campos de tensores. En este trabajo, sólo vamos a necesitar el caso más sencillo, pero vamos a explicar esta derivada con los mismos conceptos desarrollados en el marco más general.

Dada una función diferenciable $f : \mathbb{R} \rightarrow \mathbb{R}$ y $a, b \in \mathbb{R}$, su función derivada f' evaluada en $\frac{a+b}{2}$ puede ser aproximada por una derivada discreta $Df : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ definida por:

$$Df(a, b) = \frac{f(b) - f(a)}{b - a} . \quad (41)$$

Esta función es una aproximación de $f'(\frac{a+b}{2})$ de tercer orden que satisface la condición de direccionalidad:

$$Df(a, b) \cdot (b - a) = f(b) - f(a) , \quad (42)$$

y posee un límite definido

$$\lim_{b \rightarrow a} Df(a, b) = f'(a) . \quad (43)$$

Consideremos ahora el caso que más nos interesa a nosotros, el de una función diferenciable de dos variables $g : \mathbb{R}^2 \rightarrow \mathbb{R}$. Dados dos puntos $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^2$, las derivadas *parciales discretas* de g con respecto a cada uno de sus argumentos son las funciones:

$$D_1 g : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R} , \quad D_2 g : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R} , \quad (44)$$

definidas por:

$$\begin{aligned} D_1 g(x_1, y_1, x_2, y_2) &= \frac{g(x_2, y_1) - g(x_1, y_1) + g(x_2, y_2) - g(x_1, y_2)}{2(x_2 - x_1)} , \\ D_2 g(x_1, y_1, x_2, y_2) &= \frac{g(x_1, y_2) - g(x_1, y_1) + g(x_2, y_2) - g(x_2, y_1)}{2(y_2 - y_1)} . \end{aligned} \quad (45)$$

Estas dos funciones son aproximaciones de tercer orden de las correspondientes derivadas parciales en $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$ y satisfacen la condición de direccionalidad:

$$D_1 g(x_1, y_1, x_2, y_2)(x_2 - x_1) + D_2 g(x_1, y_1, x_2, y_2)(y_2 - y_1) = g(x_2, y_2) - g(x_1, y_1) . \quad (46)$$

Además, están definidas en los límites,

$$\begin{aligned} \lim_{x_2 \rightarrow x_1} D_1 g(x_1, y_1, x_2, y_2) &= \frac{1}{2} \left(\frac{\partial g}{\partial x}(x_1, y_1) + \frac{\partial g}{\partial x}(x_1, y_2) \right) , \\ \lim_{y_2 \rightarrow y_1} D_2 g(x_1, y_1, x_2, y_2) &= \frac{1}{2} \left(\frac{\partial g}{\partial y}(x_1, y_1) + \frac{\partial g}{\partial y}(x_2, y_1) \right) . \end{aligned} \quad (47)$$

3.2.2. Variables del método numérico

Apoyándonos en esta construcción de la derivada discreta, vamos a añadir dos nuevas variables a nuestra discretización. Más tarde entenderemos por qué las necesitamos. Vamos a crear una nueva densidad de energía $e_n^h \in \mathcal{V}^h$ y una nueva inversa de la temperatura $\omega_*^h \in \mathcal{V}^h$. Debemos recordar que la temperatura se define en termodinámica como una derivada de la entropía:

$$\frac{\partial \hat{s}}{\partial e} = \frac{1}{T} = \omega \quad (48)$$

Definimos entonces la nueva variable e_n^h como la proyección L_2 de la densidad de energía y la variable ω_*^h como la proyección L_2 de la derivada discreta $D_e \hat{s}$ sobre el espacio de elementos finitos:

$$\begin{aligned} \langle e_n^h, \nu^h \rangle &= \langle \hat{e}(\phi_n^h, \omega_n^h), \nu^h \rangle , \\ \langle \omega_*^h, \eta^h \rangle &= \langle D_e \tilde{s}(\phi_n^h, e_n^h, \phi_{n+1}^h, e_{n+1}^h), \eta^h \rangle , \end{aligned} \quad (49)$$

para cualesquiera funciones de forma ν^h y η^h en \mathcal{V}^h .

En definitiva, en vez de contar con dos variables de estado, trabajaremos con cuatro. La discretización de las ecuaciones de evolución (38) que proponemos es:

$$\begin{aligned} \left\langle \frac{e_{n+1}^h - e_n^h}{\Delta t}, \psi^h \right\rangle &= \langle M_T \nabla \omega_*^h, \nabla \psi^h \rangle , \\ \left\langle \tau \frac{\phi_{n+1}^h - \phi_n^h}{\Delta t}, \mu^h \right\rangle &= \langle D_\phi \tilde{s}(\phi_n^h, e_n^h, \phi_{n+1}^h, e_{n+1}^h), \mu^h \rangle - \left\langle \epsilon^2 \nabla \phi_{n+\frac{1}{2}}^h, \nabla \mu^h \right\rangle . \end{aligned} \quad (50)$$

Definimos la energía y la entropía totales en el instante t_n de una solución discreta (ϕ_n, ω_n) respectivamente como sugiere la definición (6):

$$E_n = \int_V \hat{e}(\phi_n^h, \omega_n^h) dv , \quad S_n = \int_V \left(\hat{s}(\phi_n^h, \omega_n^h) - \frac{\epsilon^2}{2} |\nabla \phi_n^h|^2 \right) dv . \quad (51)$$

En el siguiente teorema encontramos la razón que nos ha empujado a introducir dos nuevas variables en nuestro método numérico.

Teorema 1. *La discretización espacio-temporal caracterizada por las ecuaciones (49) y (49) es una aproximación de segundo orden en el tiempo cuya solución preserva las dos leyes de la termodinámica, esto es,*

$$E_{n+1} = E_n, \quad S_{n+1} \geq S_n. \quad (52)$$

Demostración. La precisión del método se deduce directamente del orden de precisión del operador diferencial discreto utilizado.

Para demostrar la conservación de la energía, basta con elegir ψ^h como función constante en la ecuación (50)₁. En efecto, como dicha ecuación debe cumplirse sea cual sea ψ^h , si la sustituimos por una constante, el término derecho de la ecuación variacional se anula, por lo que la parte izquierda debe ser nula y la energía constante.

A continuación, para demostrar la validez del segundo principio, escogemos $\mu^h = \phi_{n+1}^h - \phi_n^h$ en (50)₂ y obtenemos:

$$\begin{aligned} \tau \frac{\|\phi_{n+1}^h - \phi_n^h\|^2}{\Delta t} &= \langle D_\phi \hat{s}, \phi_{n+1}^h - \phi_n^h \rangle - \left\langle \frac{\epsilon^2}{2} (\nabla \phi_{n+1}^h + \nabla \phi_n^h), \nabla \phi_{n+1}^h - \nabla \phi_n^h \right\rangle \\ &= \langle D_\phi \hat{s}, \phi_{n+1}^h - \phi_n^h \rangle - \frac{\epsilon^2}{2} (\|\nabla \phi_{n+1}^h\|^2 - \|\nabla \phi_n^h\|^2) \\ &\quad + \langle D_e \tilde{s}, e_{n+1}^h - e_n^h \rangle - \langle D_e \tilde{s}, e_{n+1}^h - e_n^h \rangle \\ &= S_{n+1} - S_n - \langle D_e \tilde{s}, e_{n+1}^h - e_n^h \rangle. \end{aligned} \quad (53)$$

El último término de la parte derecha puede ser evaluado utilizando la proyección (49)₂ y la ecuación de evolución (50)₁:

$$\langle D_e \tilde{s}, e_{n+1}^h - e_n^h \rangle = \langle \omega_*^h, e_{n+1}^h - e_n^h \rangle = \Delta t \langle \nabla \omega_*^h, M_T \nabla \omega_*^h \rangle, \quad (54)$$

resultado siempre no negativo. Por lo tanto, (54) nos permite concluir:

$$S_{n+1} - S_n = \tau \frac{\|\phi_{n+1}^h - \phi_n^h\|^2}{\Delta t} + \Delta t \langle \nabla \omega_*^h, M_T \nabla \omega_*^h \rangle \geq 0, \quad (55)$$

que es lo que intentábamos demostrar. \square

Al principio hemos recordado el método estándar para discretizar el sistema de ecuaciones de evolución formado por (5) y (10) cuya precisión es de segundo orden. Sin embargo esa discretización no garantiza que la entropía crezca y que la energía total se conserve en cada paso de tiempo. En lugar de utilizar la regla del punto medio estándar en el tiempo, la hemos modificado introduciendo la derivada discreta definida en [3]. Además, a las dos variables de estado ω_{n+1}^h y ϕ_{n+1}^h , hemos añadido $\omega_{*(n+1)}^h$ y e_{n+1}^h como variables. Proyectando la derivada discreta de la entropía en el espacio de elementos finitos, definimos $\omega_{*(n+1)}^h$ y proyectando de la misma manera la energía interna hallamos e_{n+1}^h .

Aunque pueda parecer que esta discretización va a necesitar tiempos de cálculo el doble de grandes que la discretización estándar porque pasamos de resolver un sistema de dos ecuaciones a otro con cuatro ecuaciones, esto no es correcto. Porque trabajamos con dos ecuaciones de proyección que son sumamente simples.

4. Implementación y simulaciones numéricas

El primer apartado nos ha permitido demostrar que las ecuaciones de evolución (5) y (10) cumplen el primer y el segundo principios de la termodinámica. Después hemos presentado una discretización de las ecuaciones de evolución y hemos demostrado que en cada paso de tiempo satisface los dos primeros principios de la termodinámica. A continuación, vamos a simular el sistema de ecuaciones. En primer lugar, comentaremos el programa que hemos utilizado para las simulaciones y las librerías que hemos elegido para que el programa corra en paralelo. A continuación, terminaremos mostrando una simulación y comprobando que mientras la discretización estándar no cumple las leyes termodinámicas, nuestra discretización sí lo hace.

4.1. Implementación

4.1.1. IRIS, un código de elementos finitos

El código IRIS para investigación de elementos finitos, desarrollado y mantenido por el profesor Ignacio Romero [16], nos ha permitido simular el sistema. Es un código de elementos finitos de propósito general escrito en C++. Está bastante compartimentado, pensado para facilitar la creación de nuevos módulos, ya sean integradores, métodos algebraicos de resolución de sistemas de ecuaciones, elementos o materiales. El código puede resolver tanto problemas lineales como no lineales y contiene:

- Integradores dinámicos y cuasi-estáticos,
- Elementos sólidos tanto para pequeñas deformaciones como para deformaciones finitas,
- Elementos fluidos para las ecuaciones de Stokes y de Navier-Stokes, junto a elementos estabilizadores,
- Elementos térmicos para la ecuación de Fourier,
- Elementos acústicos para la ecuación de Helmholtz,
- Modelos de materiales lineales y no lineales elásticos, elastoplásticos, térmicos y fluidos.

Utiliza las siguientes librerías externas:

- La librería SuperLU para la solución de sistemas de ecuaciones grandes, dispersos y no simétricos,
- El proyecto LAPACK para la multiplicación, la descomposición espectral y el cálculo de normas de matrices entre otros,
- El paquete ARPACK para el cálculo de autovalores y autovectores,
- La librería Sparskit para manipular grandes matrices dispersas.

IRIS está también programado para utilizar hilos y sacar provecho de la programación paralela en máquinas con procesadores de varios núcleos.

Concretamente, para simular el sistema que hemos descrito en el apartado 2 y poder comparar la discretización que hemos propuesto en este trabajo con respecto a la estándar, hemos programado dos nuevos elementos y dos nuevos materiales. El primer elemento es de tipo phase-field y contiene no sólo las ecuaciones de evolución (5) y (10) y las de proyección (49), además alberga toda la información de la matriz tangente. El otro elemento que hemos implementado está representado únicamente por las dos ecuaciones de evolución (5) y (10) y nos ha permitido comparar los resultados. Junto a esos elementos, hemos programado dos materiales. Ambos recopilan todas las constantes físicas que caracterizan el comportamiento del material concreto que hemos simulado, así como las ecuaciones que caracterizan la expresión de la energía interna (las ecuaciones (12) y (13)) y de la entropía (las ecuaciones (21), (18), (19), (22) y (23)). Sin embargo, el primer material además calcula las derivadas discretas parciales de la energía y la entropía, mientras que el segundo calcula sus derivadas parciales analíticas.

4.1.2. Paralelización del código

Como ya hemos explicado anteriormente, la discretización espacial de las ecuaciones que queremos simular se ha llevado a cabo con elementos finitos. El método de los elementos finitos se divide en tres partes:

- El pre-proceso, etapa durante la cual se crea el material, se malla el objeto que se estudia, se crean los elementos y se ensamblan,
- La resolución del sistema lineal de ecuaciones,
- El post-proceso, fase durante la cual se representan los resultados.

Ciertamente cuando se ejecutan dos simulaciones donde la única diferencia es el tamaño de la malla, el tiempo de pre-proceso no es el mismo. No obstante, esa es una etapa que se ejecuta una sola vez a lo largo del programa. Por el contrario, en cada paso de tiempo tenemos que resolver el sistema de ecuaciones. Dicho sistema tiene una dimensión aproximadamente proporcional al cubo del número de nodos que ha generado la malla. Se comprueba fácilmente, que si la malla de una de las simulaciones es mucho más fina que la otra, el tiempo que va a invertir el programa en la etapa de proceso va a ser considerablemente mayor. En otras palabras, en sistemas suficientemente grandes, el proceso es la etapa que más tiempo tarda y cuanto mayor es el tamaño de malla, más tiempo dura la fase de proceso.

En el estudio de la solidificación, fenómeno físico que estamos estudiando, es importante que el tamaño de la interfase sea muy pequeño con respecto al volumen que se simula. Esto significa que para simular adecuadamente la solidificación necesitamos una malla extremadamente fina. En definitiva, si queremos ser capaces de simular un volumen de tamaño físicamente razonable, en un tiempo asumible, debemos recurrir a centros de computación paralela.

En esta sección, vamos a describir las librerías y modificaciones que hemos introducido en IRIS para poder correr simulaciones en entornos distribuidos.

PETSc PETSc es una librería de resolución de grandes sistemas de ecuaciones. Incluye toda serie de métodos para resolver EDPs, como métodos de subespacios de Krylov, preconditionadores, integradores, solvers, funciones para manipular matrices dispersas e incluso matrices y vectores distribuidos. Esta librería integra ya la posibilidad de ser ejecutada en paralelo con el estándar MPI. Además de implementar varios métodos de resolución de sistemas de ecuaciones tanto lineales como no-lineales, puede ser utilizada como interfaz de otras librerías de resolución de sistemas de ecuaciones tales como MUMPS, SPOOLES, SuperLU y SuperLU distribuida.

IRIS ya ofrecía la opción de utilizar sus propios métodos de resolución de sistemas de ecuaciones además de algunos otros que posee PETSc. Esencialmente, hemos tenido que programar una nueva clase ensamblador, una interfaz con los solvers paralelos de PETSc y una nueva interfaz con la clase de matriz distribuida que propone PETSc.

Recordemos que el método de los elementos finitos aproxima un sistema de ecuaciones diferenciales en otro de ecuaciones lineales que podemos escribir matricialmente de la forma:

$$Kd = f \quad (56)$$

donde el vector d es el vector cuyo valor queremos encontrar, llamado vector de desplazamientos, K es conocida por matriz de rigidez y el vector f se llama vector de fuerzas. Se suele también definir el residuo R y su matriz tangente T , esto es su derivada:

$$\begin{aligned} R &= Kd - f \\ T &= D_d R \end{aligned} \quad (57)$$

El ensamblador se encarga de ensamblar la matrices de masas, de rigidez y tangente. Como en la implementación paralela, cada proceso almacena y calcula una parte del vector de desplazamientos, del vector de fuerzas y de las matrices anteriores, es necesario que el ensamblador sepa qué ha sido ensamblado, qué no y por qué proceso, antes de formar las matrices y los vectores globales del sistema.

Además de escribir un nuevo ensamblador, hemos programado una nueva clase de matrices que sirve de interfaz entre las matrices distribuidas de PETSc y las matrices propias que posee IRIS.

Por último, hemos implementado una interfaz con varios solvers de cálculo paralelo, algunos implementados en PETSc y otros externos, pero todos se pueden utilizar a través de la propia PETSc.

En estos momentos somos capaces de recurrir a los siguientes solvers para la resolución de sistemas dispersos de ecuaciones lineales mediante computación distribuida: MUMPS, SPOOLES y SuperLU distribuida. Además, podemos utilizar los solvers propios a PETSc.

En los cálculos preliminares, hemos lanzado varias simulaciones pequeñas con uno y hasta con dieciséis procesos. Hemos observado que la librería MUMPS es la más rápida.

Validación de IRIS distribuido A la hora de validar cualquier modificación introducida en IRIS, disponemos de un árbol de benchmarks que comprueba sistemáticamente el correcto funcionamiento de las principales funciones del

programa. Para comprobar que efectivamente el modo distribuido funcionaba, tuvimos que crear nuevos benchmarks.

En primer lugar, se lanzan los mismos benchmarks que verifican el funcionamiento correcto de IRIS en modo serie, a continuación se repiten otra vez algunos de esos benchmarks, pero esta vez en modo distribuido con un sólo proceso. Por último se lanzan los benchmarks propios del modo distribuido. Estos consisten en varias simulaciones de casos mecánicos utilizando los distintos solvers que hemos mencionado antes (MUMPS, SPOOLES y SuperLU distribuida) y distinto número de procesos (1, 2 y 7). Al final se comparan los resultados y las fuerzas obtenidos en unos nodos concretos con respecto a una solución de referencia.

En la figura 2, se ha representado el tiempo total de ejecución de un benchmark en función del número de procesos. Se resuelve un problema estático y lineal de mecánica del sólido sencillo de resolver. Se observa que al aumentar el número de procesos el tiempo de ejecución disminuye, como era de esperar. Cuando alcanza cierto número de procesos, el tiempo no disminuye pudiendo incluso aumentar. Este fenómeno es debido a la gestión de cada proceso que el sistema operativo debe mantener.

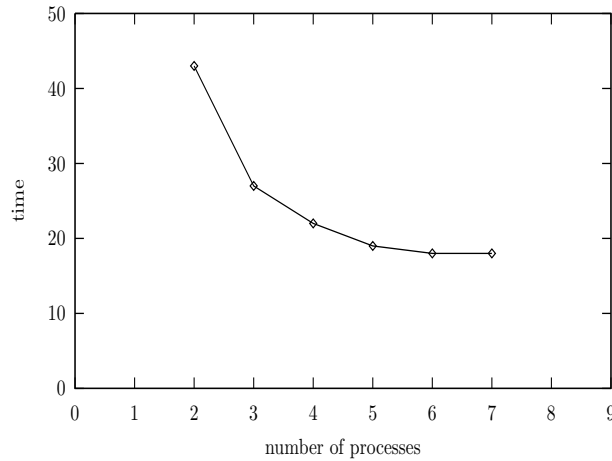


Figura 2: Tiempo de ejecución en función del número de procesos.

4.2. Simulaciones y resultados

Hemos realizado simulaciones muy parecidas a las presentadas en [18]. En dicho artículo comentan distintas simulaciones del modelo I de Wang, el mismo que hemos descrito en este trabajo. Ese trabajo nos ha permitido contrastar nuestros resultados y comprobar que eran razonables.

4.2.1. Procesos de subenfriamiento

Hemos simulado la solidificación de un cuadrado (2D) de níquel con una fase sólida y otra líquida subenfriada. La fase sólida se encuentra a la temperatura de solidificación T_M y la líquida se encuentra a una temperatura más fría $T_0 = T_M - 210$. Este caso puede parecer extraño a primera vista, pero tiene su

importancia. La termodinámica de los cambios de fase demuestra que cuando se produce una solidificación, tanto el líquido como el sólido deben permanecer ambos a la temperatura T_m , mientras coexistan las dos fases. En la figura 3, se ha representado un diagrama de enfriamiento. Se muestra la evolución de la temperatura de un cuerpo durante su enfriamiento a lo largo del tiempo. Inicialmente, el cuerpo se encuentra completamente en estado líquido a una temperatura elevada, es decir, superior a la de solidificación. Se deja enfriar y cuando alcanza la temperatura de solidificación T_m , el cambio de fase comienza. El líquido empieza a transformarse en sólido y mientras coexisten las dos fases la temperatura es constante e igual a T_m . Cuando el cuerpo es totalmente sólido, sigue enfriándose hasta temperatura ambiente.

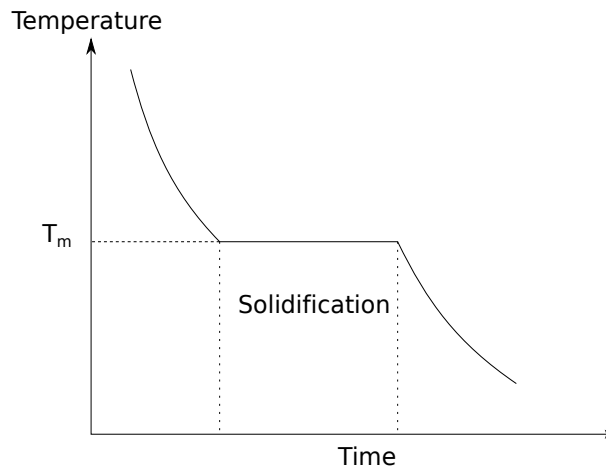


Figura 3: Evolución de la temperatura durante un enfriamiento lento y en equilibrio.

No obstante, la termodinámica estudia los estados de equilibrio, por lo que no significa que sea imposible encontrar el sólido a una temperatura distinta a la del líquido durante el cambio de fase. De hecho, la teoría de la solidificación estable señala que el líquido solidifica formando una esfera, que se va agrandando a lo largo del tiempo. El aumento de volumen de la esfera no es más que la conversión del líquido en sólido.

Sin embargo, cuando se obtienen fundiciones o aceros por el proceso industrial del temple, se observan en algunas ocasiones estructuras dendríticas en las piezas. La importancia de estas estructuras reside en las propiedades que confieren, distintas a las propiedades de las piezas sin dendritas. En la figura 4, podemos apreciar una estructura no homogénea en la pieza. Se observan distintas ramas que siguen todas una dirección. Esta estructura es típica de un proceso de no equilibrio.

Durante muchos procesos industriales, se enfría el líquido rápidamente por debajo de su temperatura de solidificación. En la figura 5 se puede ver un diagrama de subenfriamiento. Se representa la evolución de la temperatura en función del tiempo. Como se ha enfriado el líquido muy rápidamente, se sigue enfriando sin que empiece el cambio de fase por debajo de T_m . Cuando el sólido empieza a formarse, se desprende calor y la mezcla se calienta hasta que vuelve a alcanzar la temperatura de cambio de fase T_m . En ese momento el cambio de fase

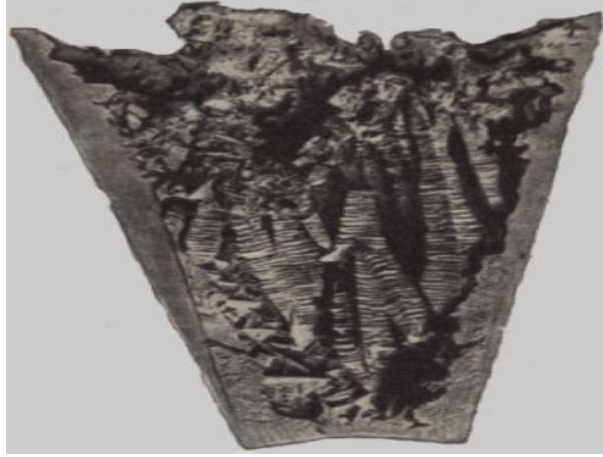


Figura 4: Ejemplo de una pieza con una cristalización dendrítica. Fuente [1]

prosigue a temperatura constante hasta que la transformación se completa. De nuevo, cuando ya todo el cuerpo es sólido, su enfriamiento normal prosigue.

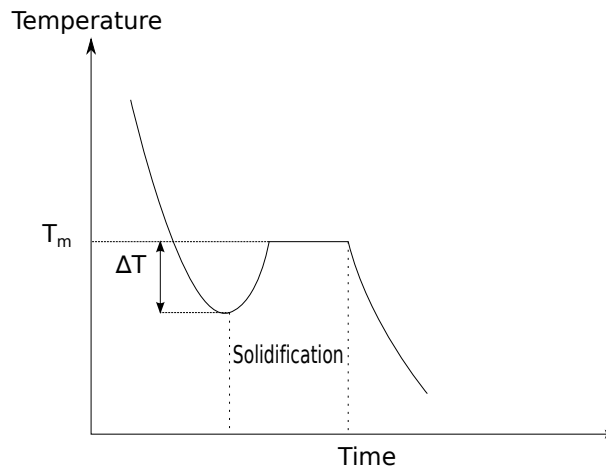


Figura 5: Proceso de subenfriamiento.

Queremos simular precisamente este tipo de transformaciones de no equilibrio, por lo que esperamos encontrar en nuestras simulaciones estructuras dendríticas.

4.2.2. Simulaciones

Las propiedades físicas del níquel que hemos utilizado son las mismas que en [18]. La situación inicial en la que se encuentra el cuadrado de níquel que hemos simulado aparece reflejada en las figuras 6 y 7. Inicialmente, la gran parte del níquel está en estado líquido (en rojo en la figura 6). Sólo una pequeña zona en forma de elipse se encuentra en estado sólido. Podemos apreciar que hemos regularizado la transición de fase. En lugar de tener una transición abrupta en

la que un nodo y su vecino se encuentren en fases distintas, hemos impuesto una transición progresiva. En la figura 7, vemos que el sólido se encuentra inicialmente a la temperatura T_0 (en rojo), mientras que el sólido está a la temperatura de cristalización (en azul). La temperatura también está regularizada y hay una pequeña transición entre la temperatura del sólido y la del líquido. La malla que hemos considerado en todas las simulaciones ha sido 200×200 . Esta malla nos ha parecido conveniente porque, por un lado, no ha necesitado tiempos de cálculo desorbitados y por otro, ha representado un tamaño de la interfase suficientemente fino.

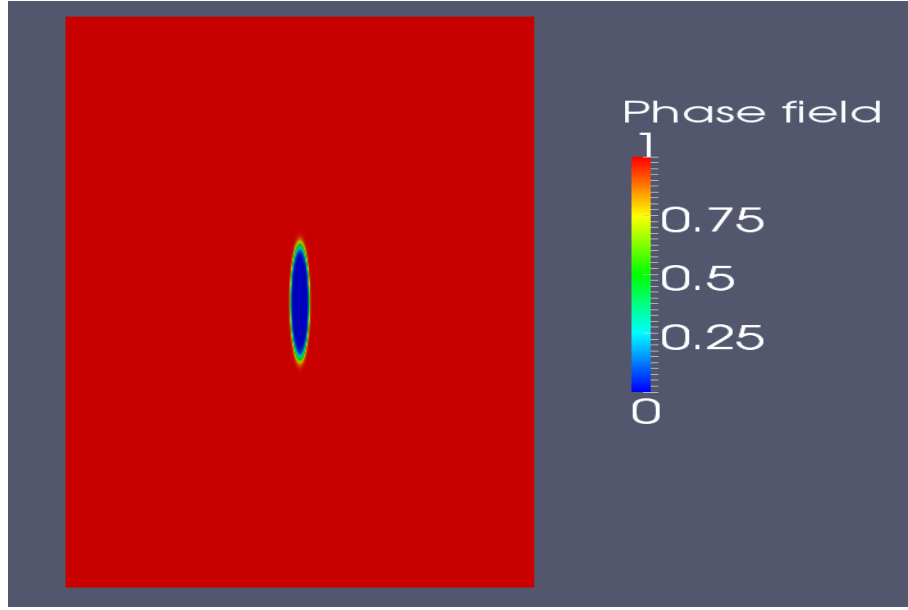


Figura 6: Estado inicial del campo de fase.

En las figuras 8 y 9, se representan el campo de fase y la variable ω en un instante intermedio. Si se comparan con las figuras anteriores, 6 y 7, se observa que el sólido se ha extendido en la dirección (Oy), y además han aparecido unas pequeñas ramificaciones en dirección perpendicular (Ox).

Si dejamos evolucionar un poco el sistema y nos fijamos en un instante posterior (figuras 10 y 11), apreciamos el aumento del número de ramificaciones con respecto a la situación anterior. Además, las ramificaciones anteriores se han extendido y parecen ya auténticas dendritas.

Por último, en las figuras 12 y 13, encontramos el estado del campo de fase y de w , al final de la simulación, al cabo de $t = 2$ unidades de tiempo con respecto al inicio. Se ha formado una estructura dendrítica sin ningún género de dudas. En lugar de obtener un eje principal (eje dendrítico) y dendritas más pequeñas perpendiculares, hemos acabado con un eje dendrítico, dendritas perpendiculares a él y otras dendritas que nacen a lo largo de las anteriores. En el artículo [18], mencionan que, para determinados valores de los parámetros de la simulación, las propias dendritas se ramifican a su vez. Concluimos que nuestro resultado no contradice los resultados de las simulaciones en [18], sino todo lo contrario.

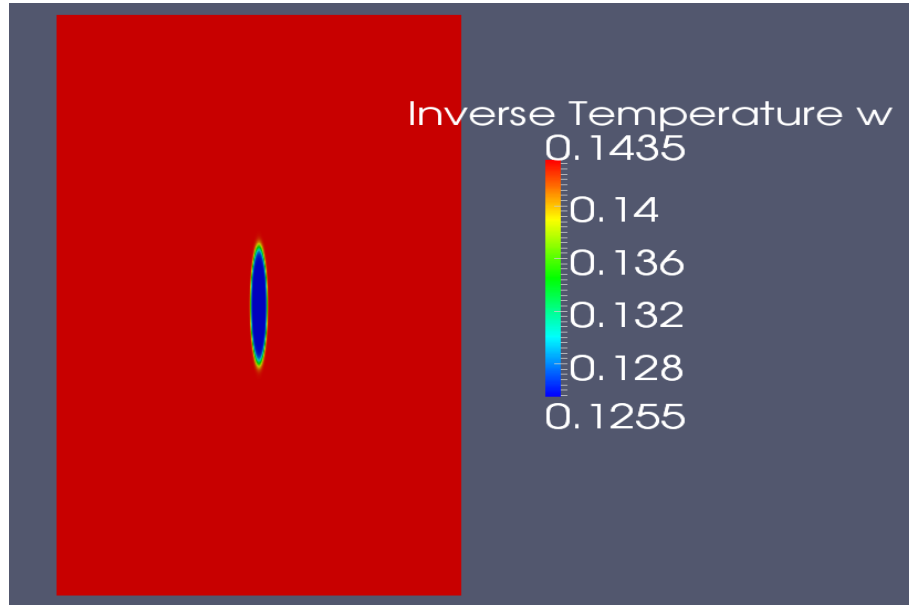


Figura 7: Estado inicial del inverso de la temperatura ω

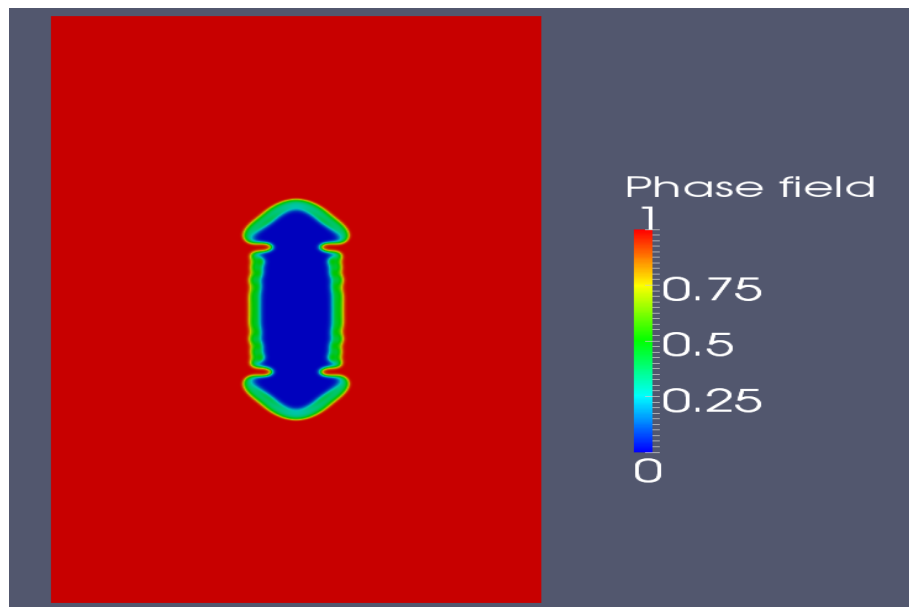


Figura 8: Estado del campo de fase en el instante $t = 0,031$.

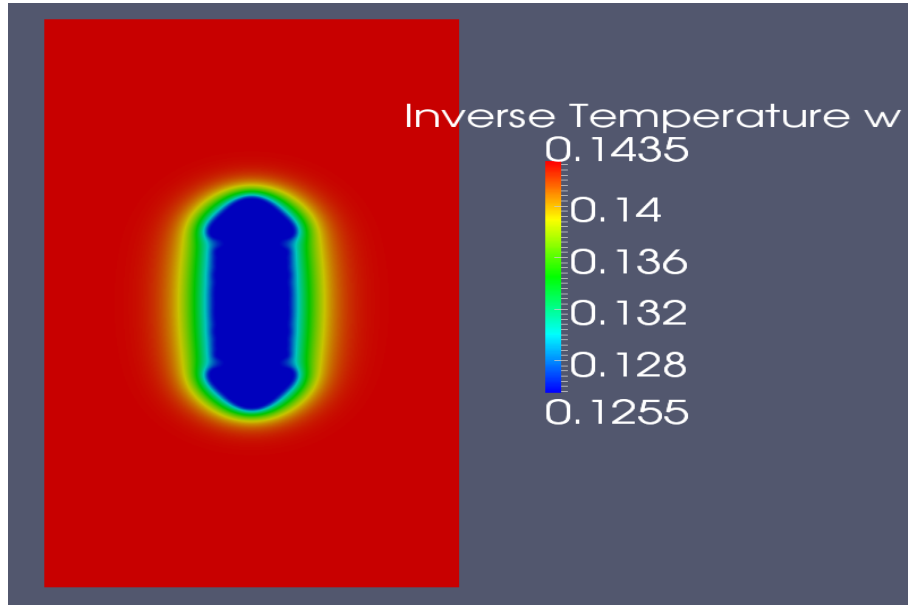


Figura 9: Estado del del inverso de la temperatura w en el instante $t = 0,031$



Figura 10: Estado del campo de fase en el instante $t = 0,06$.

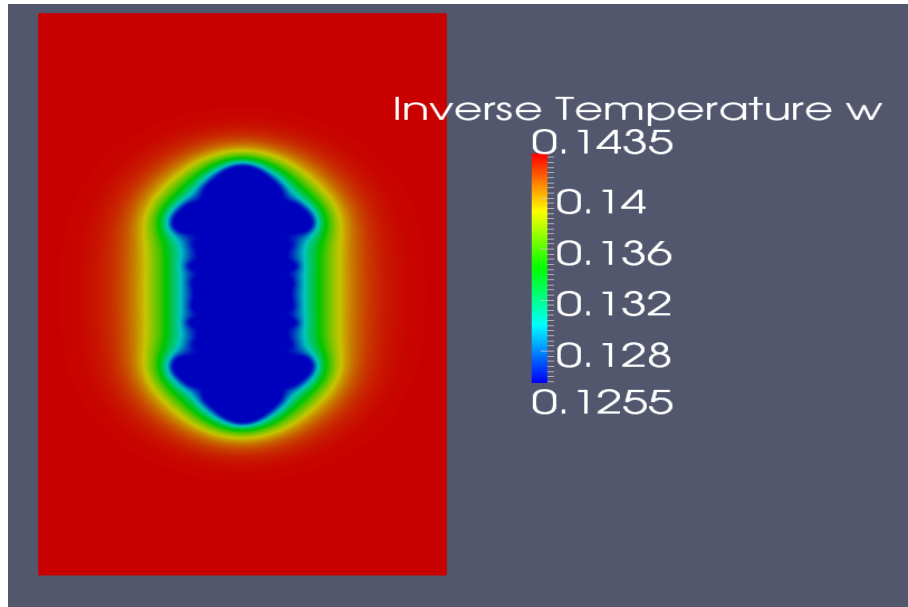


Figura 11: Estado del del inverso de la temperatura w en el instante $t = 0,06$

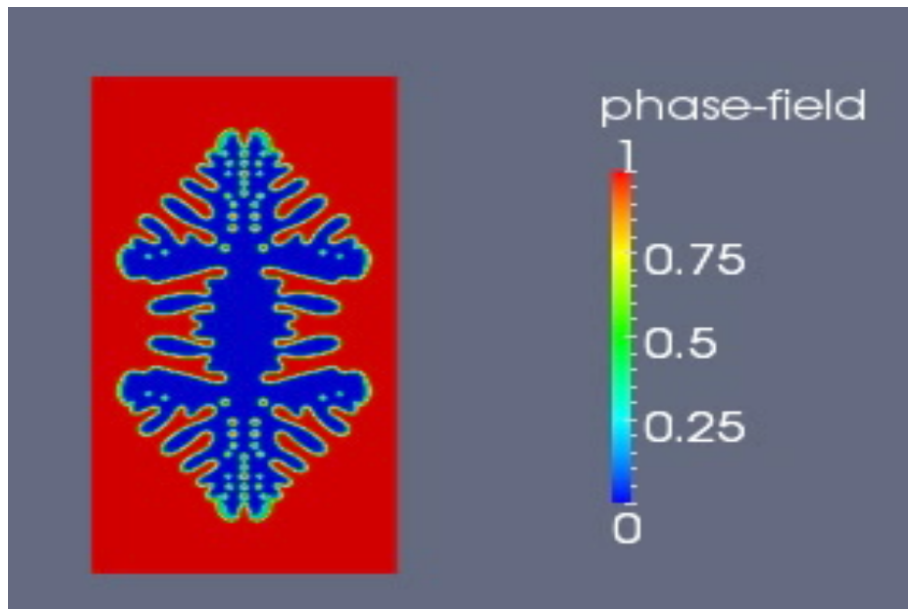


Figura 12: Evolución del campo de fase de un cuadrado de níquel subenfriado en el instante $t = 0,2$

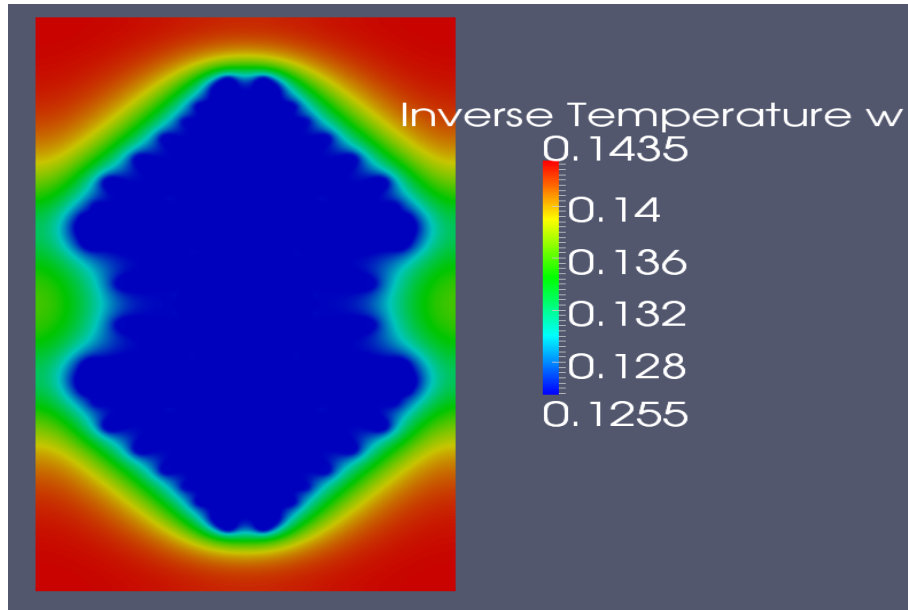


Figura 13: Evolución del inverso de la temperatura w de un cuadrado de níquel subenfriado en el instante $t = 0,2$

Por último, debemos verificar que nuestra simulación recoge la consistencia termodinámica de la discretización que hemos propuesto. Por ello, también hemos simulado el sistema anterior con la discretización estándar, esto es, elementos finitos en el espacio y la fórmula del punto medio en el tiempo. Visualmente no se aprecian diferencias ni en la evolución del campo de fase, ni en la evolución de w entre la simulación con nuestra discretización y la estándar.

También hemos representado la evolución de la energía (figuras 15 y 14) y de la entropía (figuras 17 y 16) totales de ambos sistemas. El teorema 1, página 16 garantiza que nuestra discretización satisface el primer principio, es decir, la energía total es constante, y el segundo, esto es, la entropía es no decreciente. En la figura 14 comprobamos que efectivamente la energía del sistema con la discretización propuesta posee la misma energía total en todos los instantes de tiempo. En la figura 16 también se verifica que la entropía del sistema no decrece cuando el tiempo avanza.

Por el contrario, si nos fijamos ahora en las figuras 15 y 17, apreciamos que a pesar de que la entropía parece no decrecer en la simulación de la discretización estándar, la energía no permanece constante.

El método presentado e ilustrado en los trabajos [13, 14, 15], permite generar discretizaciones termodinámicamente consistentes de problemas que se puedan formular de una determinada manera. En este trabajo, queríamos aplicar ese método a un nuevo sistema que no fuera mecánico. Elegimos la solidificación porque acopla un campo de fase con la temperatura. En los apartados anteriores ya extraímos las ecuaciones de evolución que rigen la solidificación y desarrollamos una discretización que satisfacía las leyes de la termodinámica. Además, deseábamos simular el sistema. El programa de investigación de elementos finitos, IRIS, fue el punto de partida. Programamos nuevos materiales y elementos

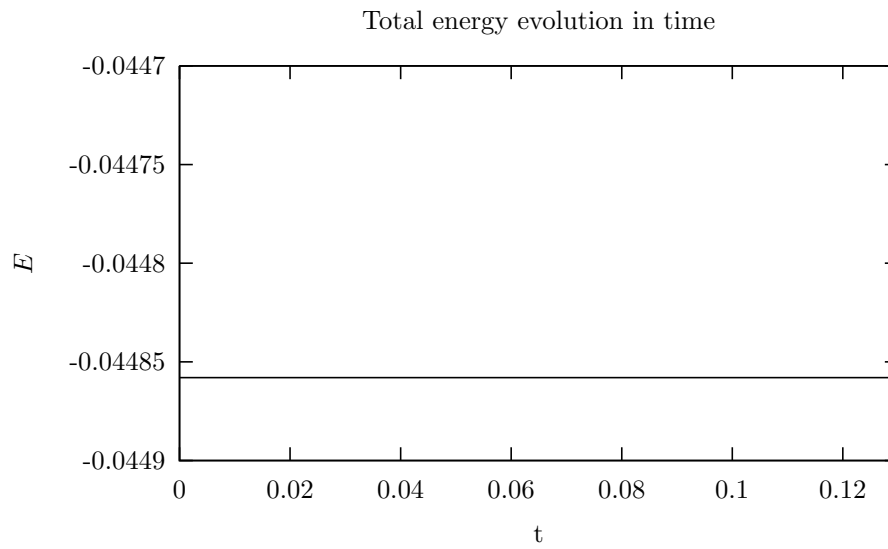


Figura 14: Evolución de la energía con la discretización TC

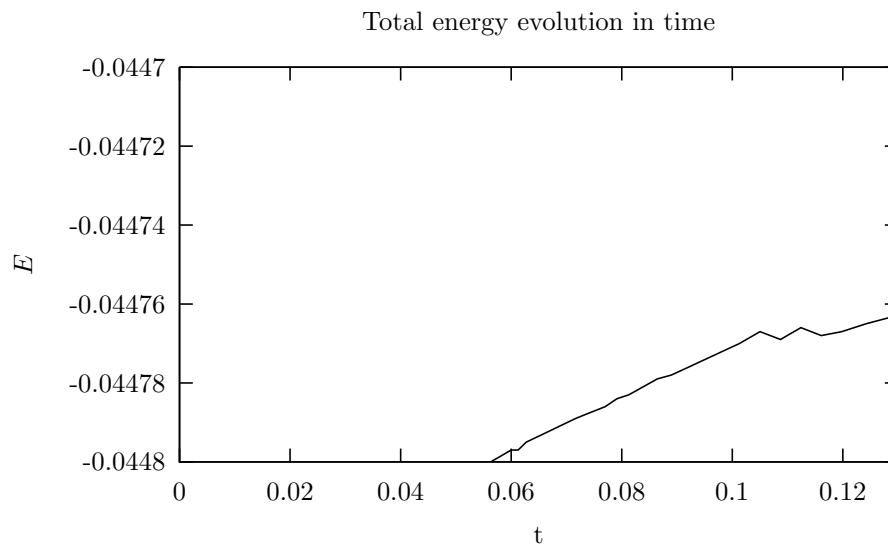


Figura 15: Evolución de la energía con la regla del punto medio

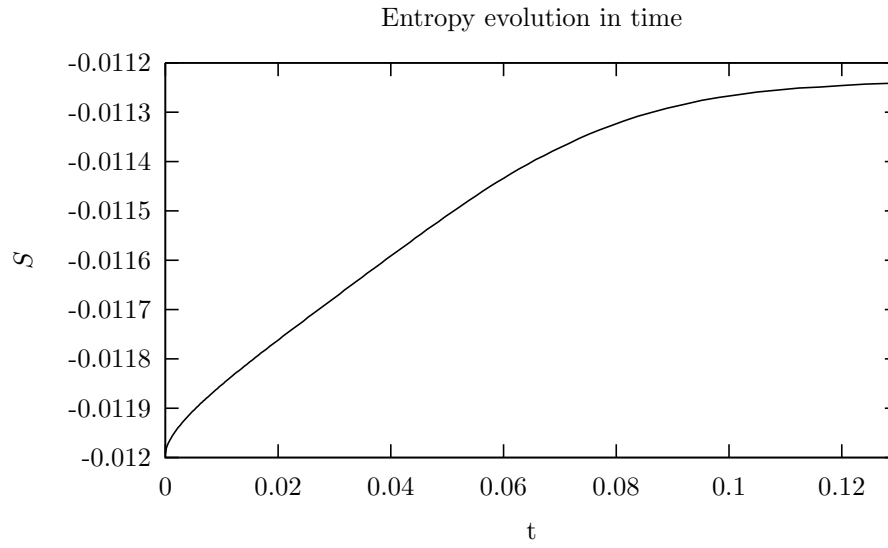


Figura 16: Evolución de la entropía con la discretización TC

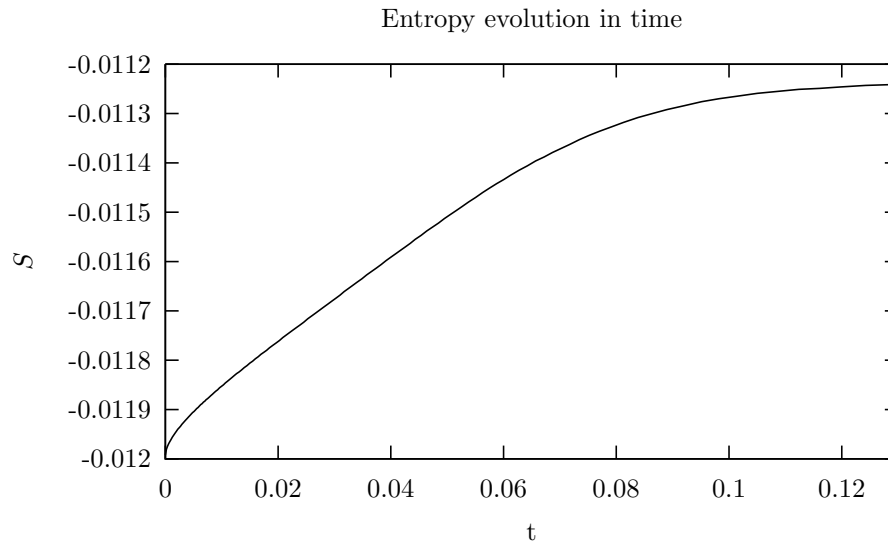


Figura 17: Evolución de la entropía con la regla del punto medio

que contuvieran toda la información física y las ecuaciones del sistema. Como la física de la solidificación nos obligaba a trabajar con mallas muy finas, implementamos en IRIS la posibilidad de correr simulaciones en paralelo, a través de la librería PETSc. Esta librería emplea MPI como paradigma de programación distribuida. Tuvimos que programar una nueva clase ensamblador y varias interfaces para sacar provecho a las posibilidades que ofrece PETSc. Por último, creamos diversos benchmarks que nos permitieron validar los resultados obtenidos con la versión distribuida de IRIS.

Una vez el programa a punto, simulamos un cuadrado de Níquel con una pequeña región elíptica de sólido a temperatura de cristalización y una gran cantidad de líquido subenfriado. Obtuvimos unos resultados que ya fueron recogidos por otros autores en [18]. Finalmente, comprobamos que, mientras la energía interna de la discretización estándar no permanecía constante en el proceso, la novedosa discretización que desarrollamos sí mostraba una energía interna constante y una entropía no decreciente, a lo largo de todo el proceso de solidificación simulado.

5. Conclusión

A la hora de modelizar transiciones de fase, los métodos de tipo phase-field han demostrado ser una alternativa convincente a los métodos "sharp interfase". Sin embargo, es importante que las transiciones de fase respeten las leyes de la termodinámica. Hasta ahora, los métodos numéricos que se han utilizado para simular estos sistemas no garantizan la consistencia termodinámica en cada punto de integración. En este trabajo hemos presentado un método numérico basado en el marco GENERIC que asegura la satisfacción de las leyes termodinámicas.

Primero, dedujimos las ecuaciones de evolución que rigen la solidificación de sustancias puras, a partir de los dos principios termodinámicos y unas pocas hipótesis. A continuación, las discretizamos con la fórmula del punto medio en el tiempo y elementos finitos en el espacio. No obstante, las ecuaciones discretizadas no conservan la energía interna y nada impide el decrecimiento de la entropía del sistema que modelan. Para solucionar esta circunstancia, añadimos dos nuevas variables a las variables de estado, modificamos la fórmula del punto medio sustituyendo las derivadas parciales por derivadas discretas, obtuvimos la forma débil discretizada del sistema de ecuaciones y demostramos la consistencia termodinámica de la discretización final.

Además, tomamos como base el programa de investigación de elementos finitos IRIS, desarrollado en C++ por el catedrático Ignacio Romero, al que añadimos algunas clases nuevas. Estas clases contenían toda la información no sólo del sistema de ecuaciones de evolución, sino también de las propiedades físicas del material que simulamos. Utilizamos por otra parte la librería PETSc para resolver el sistema de ecuaciones en paralelo. Estas modificaciones nos permitieron correr simulaciones donde la energía se conserva y la entropía crece en el caso de nuestra discretización propuesta. Por el contrario, cuando simulamos el mismo sistema, pero esta vez discretizado con la regla del punto medio, la energía interna no se conservó, contradiciendo entonces el primer principio de la termodinámica.

Este trabajo aplica el formalismo presentado en [13, 14, 15] a un sistema

phase-field. De esta manera, mostramos el gran abanico de problemas donde esta metodología puede ser utilizada.

En breve, vamos a emplear Magerit, el superordenador de la UPM, para simular la discretización propuesta. Esperamos poder simular problemas 3D con un tamaño de malla suficientemente fino para captar la física del proceso de solidificación. Debemos mencionar no obstante, que gran parte del algoritmo es secuencial. En efecto, en primer se debe resolver un gran sistema matricial construido mediante elementos finitos a cada paso de tiempo. Y la integración en el tiempo sólo se puede ejecutar de forma secuencial, pues para un paso de tiempo determinado necesitamos disponer de los valores de las variables nodales en el paso de tiempo anterior. Por lo tanto, la paralelización contribuye únicamente en la disminución del tiempo de cálculo del sistema matricial, que debe resolverse en cada paso de tiempo.

Referencias

- [1] T. BRUZOS, *Curso online de la solidificación de los metales*.
<http://www.sabelotodo.org/metallurgia/cristalmetal.html>.
- [2] P. CECCHI, *Explicación online de los métodos de solidificación*.
http://www.comosehace.cl/procesos/PaulinaCecci/complemento_Solidificacion.html.
- [3] O. GONZALEZ, *Time integration and discrete hamiltonian systems*, Journal of Nonlinear Science, 6 (1996), pp. 449–467.
- [4] R. GONZALEZ-CINCA, R. FOLCH, R. BENITEZ, L. RAMIREZ-PISCINA, J. CASADEMUNT, AND A. HERNANDEZ-MACHADO, *Phase-field models in interfacial pattern formation out of equilibrium*, in condensed matter, (2003), pp. 203–236.
- [5] L. GOTUSSO, *On the energy theorem for the lagrange equations in the discrete case*, Applied Mathematics and Computation, 17 (1985), pp. 129 – 136.
- [6] T. J. R. HUGHES, *The Finite Element Method*, Standford Bookstore, Fall 1996.
- [7] T. ITOH AND K. ABE, *Hamiltonian-conserving discrete canonical equations based on variational difference quotients*, Journal of Computational Physics, 76 (1988), pp. 85 – 102.
- [8] R. A. LABUDDE AND D. GREENSPAN, *Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion. I. Motion of a single particle*, Numer. Math., 25 (1976), pp. 323–346.
- [9] ———, *Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion. II. Motion of a system of particles*, Numer. Math., 26 (1976), pp. 1–16.
- [10] G. B. MCFADDEN, A. A. WHEELER, R. J. BRAUN, S. R. CORIELL, AND R. F. SEKERKA, *Phase-field models for anisotropic interfaces*, Phys. Rev. E, 48 (1993), pp. 2016–2024.
- [11] R. I. MCLACHLAN, G. R. W. QUISPTEL, AND N. ROBIDOUX, *Geometric integration using discrete gradients*, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 1021–1045.
- [12] O. PENROSE AND P. C. FIFE, *Thermodynamically consistent models of phase-field type for the kinetic of phase transitions*, Physica D: Nonlinear Phenomena, 43 (1990), pp. 44 – 62.
- [13] I. ROMERO, *Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 706–732.
- [14] ———, *Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part i: Monolithic integrators and their application to finite strain thermoelasticity*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1841 – 1858.

- [15] ———, *Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part ii: fractional step methods*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 2235 – 2248.
- [16] ———, *IRIS: USER MANUAL*, April 2011.
- [17] S.-L. WANG, R. SEKERKA, A. WHEELER, B. MURRAY, S. CORIELL, R. BRAUN, AND G. MCFADDEN, *Thermodynamically-consistent phase-field models for solidification*, Physica D: Nonlinear Phenomena, 69 (1993), pp. 189 – 200.
- [18] A. WHEELER, B. MURRAY, AND R. SCHAEFER, *Computation of dendrites using a phase field model*, Physica D: Nonlinear Phenomena, 66 (1993), pp. 243 – 262.